DocKylin: A Large Multimodal Model for Visual Document Understanding with Efficient Visual Slimming

Jiaxin Zhang[1*], **Wentao Yang**[1*], **Songxuan Lai**[2], **Zecheng Xie**[2,✉], **Lianwen Jin**[1,✉]

[1]South China University of Technology
[2]Huawei Cloud
xiezecheng1@huawei.com, eelwjin@scut.edu.cn

Abstract

Current multimodal large language models (MLLMs) face significant challenges in visual document understanding (VDU) tasks due to the high resolution, dense text, and complex layouts typical of document images. These characteristics demand a high level of detail perception ability from MLLMs. While increasing input resolution improves detail perception capability, it also leads to longer sequences of visual tokens, increasing computational costs and straining the models' ability to handle long contexts. To address these challenges, we introduce DocKylin, a document-centric MLLM that performs visual content slimming at both the pixel and token levels, thereby reducing token sequence length in VDU scenarios. We introduce an Adaptive Pixel Slimming (APS) preprocessing module to perform pixel-level slimming, increasing the proportion of informative pixels. Moreover, we propose a novel Dynamic Token Slimming (DTS) module to conduct token-level slimming, filtering essential tokens and removing others to adaptively create a more compact visual sequence. Experiments demonstrate DocKylin's promising performance across various VDU benchmarks and the effectiveness of each component.

Introduction

With the substantial growth in data and model sizes, as well as alignment with human preferences, current Large Language Models (LLMs) (OpenAI 2022; Touvron et al. 2023; The Vicuna Team 2023; Bai et al. 2023a; The InternLM Team 2023; Yang et al. 2023) have demonstrated remarkable capabilities in reasoning, common sense understanding, expertise in specialized knowledge, zero/few-shot learning, and instruction following, shedding light on the potential for Artificial General Intelligence. Multimodal Large Language Models (MLLMs) aim to enhance these powerful LLMs by endowing them with multimodal capabilities, enabling the integration of diverse modalities beyond text alone (Yin et al. 2024; Fu et al. 2024).

Although there has been significant progress in the development of MLLMs, their performance in visual document understanding (VDU) tasks remains suboptimal (Liu et al. 2023a; Zhang et al. 2024b). A primary reason for this is their support for input resolutions of only up to 448×448, which is inadequate for high-resolution, fine-grained document images. Numerous efforts (Yu et al. 2024; Liu et al. 2024d; Li et al. 2024b; Wei et al. 2025; Bai et al. 2023b; Ye et al. 2023a; Lin et al. 2023; Liu et al. 2024a) have recently been made to increase the input resolution to address this limitation and achieved notable improvements. However, these efforts still face several challenges: **1)** Increasing the resolution also escalates the redundant regions within documents, which results in an increased number of redundant visual tokens, complicating the task for LLMs to identify the correct answers. **2)** Some methods (Li et al. 2024b; Wei et al. 2025; Liu et al. 2024a; Bai et al. 2023b) employ a fixed rectangular resolution, which can lead to text distortion given the typically large aspect ratios of document images. Moreover, scaling smaller images to larger ones results in inefficiencies. **3)** To manage the extended sequences of visual tokens resulting from high resolution, most methods (Li et al. 2024b; Yu et al. 2024; Liu et al. 2024d; Bai et al. 2023b) employ a fixed compression ratio or extract a predetermined number of tokens. While this approach may be suitable for natural scene images, it is less effective for document scenarios where content density varies significantly. For example, images of scientific papers are information-dense, whereas images of identification cards are comparatively sparse.

To tackle these challenges, we propose DocKylin, a document-centric MLLM that incorporates several innovative designs: Firstly, we propose a parameter-free Adaptive Pixel Slimming (APS) preprocessing technique that utilizes gradient information to identify and eliminate redundant regions within document images, thereby reducing the proportion of redundant pixels and enhancing computational efficiency. Secondly, we adopt a more flexible image encoding strategy by capping the maximum number of pixels instead of fixing resolutions. This allows for variable resolutions and aspect ratios. Our lightweight Swin encoder (Liu et al. 2021) processes high-resolution document images holistically, ensuring feature coherence. Furthermore, we introduce a parameter-free Dynamic Token Slimming (DTS) method based on dual-center clustering. DTS efficiently filters informative tokens from a large set of visual tokens, resulting in reduced and dynamically sized visual sequences. Incorporating these innovations, we developed our DocKylin, which demonstrates superior performance across mul-

*These authors contributed equally.

tiple VDU benchmarks. Extensive experiments demonstrate that DocKylin outperforms existing methods across various VDU benchmarks. Furthermore, both the proposed APS and DTS are parameter-free and modular, facilitating easy integration into existing MLLMs to achieve enhanced performances.

Related Works

Text-centric MLLMs for VDU

Visual document understanding (VDU) (Cui et al. 2021; Zhang et al. 2023a; Xu et al. 2020; Kim et al. 2022) centers on the automated processing, interpretation, classification, and extraction of information from document images with varied typesetting formats. This task is particularly challenging due to the diverse layouts (Cheng et al. 2023), often poor image quality (Zhang et al. 2024a, 2023b, 2022), and complex structures within the documents.

Leveraging large language models (LLMs) trained on extensive datasets, emerging MLLMs have recently demonstrated significant potential in VDU tasks. However, their performance still falls short of expectations (Liu et al. 2023a; Zhang et al. 2024b; Tang et al. 2024b; Shan et al. 2024). Therefore, there has been a surge of recent work studying how to enhance the perception, recognition, and understanding abilities of MLLMs in such challenging scenarios.

Some approaches (Tanaka et al. 2024; Kim et al. 2023; Lu et al. 2024; Liao et al. 2024) leverage additional OCR engines to recognize text in images and input it into MLLMs. Others (Li et al. 2024b; Liu et al. 2024d; Bai et al. 2023b; Dong et al. 2024; Feng et al. 2023a; Ye et al. 2023a; Wei et al. 2025; Zhao et al. 2024a,b) enhance the fine-grained perceptual capabilities of MLLMs by maintaining the high resolution of the input images. This includes strategies such as splitting the image into small pieces to accommodate the ViT's requirement for fixed and small-size inputs and utilizing visual encoders that are more suitable for text images. Authors of (Feng et al. 2023b; Wang et al. 2023) explore learning text grounding for MLLMs, while (Hu et al. 2024; Tang et al. 2024a) investigate the use of more superior and larger scale instruction data to boost performance. Although these efforts have significantly improved the performance of MLLMs on text-centric tasks, there is still substantial room for improvement.

Visual Token Compression in MLLMs

Maintaining high-resolution inputs is a straightforward and effective method to enhance the performance of MLLMs on text images. However, increased resolutions inevitably lead to a longer visual sequence. To eliminate this problem, various methods are proposed for compressing the length of visual sequences in current MLLMs. One of the earlier and commonly used approaches (Li et al. 2023; Wang et al. 2024) involves utilizing a group of learnable query tokens to extract information in a cross-attention manner. Another method (Chen et al. 2023a; Yu et al. 2024) involves concatenating adjacent tokens along the channel dimension to reduce sequence length. Additionally, convolutional neural networks have also been proven effective for visual token compression (Cha et al. 2024; Hu et al. 2024). However, all these methods either extract tokens of a fixed length or compress them at a fixed ratio, which is unsuited for document images, where the content density can vary significantly. A token sequence that is too short or a compression ratio that is too high can result in an inadequate representation. Conversely, longer token sequences or lower compression rates can reduce computational efficiency.

Recent studies (Cai et al. 2024) have shown that the optimal length of visual token sequences varies for each sample. Consequently, some recent approaches have explored the adaptive compression of visual sequences. Most of these methods (Liu et al. 2024c; Shang et al. 2024; Chen et al. 2025; Lin et al. 2024) dynamically discard nonessential tokens using attention scores between visual tokens or between visual tokens and class tokens. However, considering the differences between text and natural objects with complete semantics, this paradigm may not be suitable for text images. For text-centric images, HRVDA (Liu et al. 2024a) employs text position annotations to train a separate visual token filtering network, which requires introducing additional parameters and training procedures. Compared to the methods above, we explore alternative dynamic compression schemes specifically for document images, which include pixel-level redundancy removal and unsupervised clustering for filtering out nonessential tokens.

Methodology

As illustrated in Fig. 1, the architecture of DocKylin incorporates an image encoder, an MLP layer that aligns the visual modality to the language modality, and a LLM. Building on this framework, DocKylin introduces the following enhancements: **1)** An Adaptive Pixel Slimming (APS) module preprocesses images before they are fed into the image encoder, which can remove redundant regions and reduce resolution while increasing the proportion of informative pixels. **2)** A more lightweight and flexible image encoder by capping the maximum number of pixels instead of fixing resolutions. **3)** A Dynamic Token Slimming (DTS) module handles the visual tokens produced by the visual encoder and MLP layer. DTS further removes nonessential tokens to enhance both performance and efficiency.

In this paper, terms like 'redundant' and 'nonessential' refer to areas that can be masked without altering the document's overall meaning, allowing both humans and models to answer questions correctly. Examples include the background of the document or color blocks in bar charts.

Adaptive Pixel Slimming

Perceiving textual information in images typically requires maintaining a high input resolution, especially for text-rich document images. This poses significant challenges for current resource-intensive MLLMs. To address this issue, existing methods employ strategies such as dividing the images into patches for processing (Ye et al. 2023a), employing visual encoders that support high resolution (Wei et al. 2025), leveraging OCR results as additional inputs (Tanaka et al. 2024; Kim et al. 2023), or applying compression transformations to the images beforehand (Feng et al. 2023a).

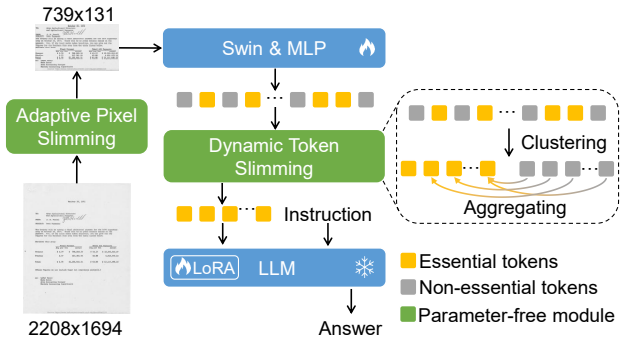

Figure 1: The overall architecture of our DocKylin model.

Here, orthogonal to the approaches mentioned above, we propose a novel approach called Adaptive Pixel Slimming (APS) to efficiently reduce the resolution of document images. APS is based on the observation that, as illustrated in Fig. 2, while document images usually exhibit high resolution, they commonly contain a significant proportion of redundant regions. These regions not only increase the computational burden on the visual encoder but may also pose challenges (Liu et al. 2024d,a) to the reasoning capabilities of the LLMs by the increased redundant visual tokens. APS is thus designed to identify and remove these regions. Specifically, as shown in Fig. 2, during the gradient extraction stage, we utilize the Sobel operator to extract the gradient map I_G from the original image I_S. Regions with low gradient values are considered smooth and devoid of significant information. Therefore, in the redundant region determination stage, we identify the redundant regions in the horizontal and vertical directions by locating areas with consistently low gradient values. For instance, to identify redundant regions in the image's horizontal direction: if the sum of gradient values for a specific row is below a predefined threshold, we classify that row as redundant. A sequence of contiguous redundant rows constitutes a redundant region in the horizontal direction. The vertical direction follows a similar procedure. The final image can be obtained by removing the identified redundant rows and columns.

Flexible and Lightweight Image Encoder

Most existing MLLMs (Li et al. 2024b) utilize a fixed rectangular resolution, such as 224×224 or 336×336. This approach distorts the original aspect ratio of the images, leading to text deformation and inefficiencies when upscaling images smaller than the preset resolution. While some methods (Ye et al. 2023a; Yu et al. 2024) allow for more flexible aspect ratios and resolutions, they often employ a patch-based processing strategy. Such a strategy may be suitable for natural scene images where the continuity and context of visual elements are less critical. However, it is less effective for document images that contain dense text (Liu et al. 2024d; Huang et al. 2024). Segmenting such images can disrupt the textual content, severing lines of text or splitting words and sentences across different patches. This discontinuity can impair the ability to accurately recognize and interpret textual information essential for effective document image understanding.

We employ a more flexible approach that can accommodate varying resolutions and aspect ratios without the need for image patching. Specifically, for an image obtained by APS with the size of $H \times W$, we set a maximum allowable number of pixels, MAX_SIZE. If the product of H and W is less than MAX_SIZE, the image will not be resized. Otherwise, a scaling factor $r = \mathrm{Int}(\sqrt{\frac{\mathrm{MAX\_SIZE}}{H \times W}})$ is calculated, and then the image is resized to $rH \times rW$. Here $\mathrm{Int}()$ denotes rounding down to the nearest integer to ensure that the resulting image does not exceed the maximum allowable pixels. One advantage of setting a maximum number of pixels rather than fixed dimensions is the ability to support more varied resolutions and preserve the original aspect ratio. Additionally, this manner avoids the loss of image details due to resolution reduction as much as possible without increasing computational resource consumption. For example, if we set the maximum pixel number MAX_SIZE as 960×960, we can accommodate an input resolution of 1280×720 without alteration. In contrast, a fixed-dimension strategy would require resizing the image to 960×960, resulting in a 25% compression of the long side. Both strategies consume comparable computational resources since the number of visual tokens remains consistent.

Although ViT pre-trained with CLIP (Radford et al. 2021) demonstrates impressive generalization capabilities for natural scene images, they perform suboptimally on dense text images (Li et al. 2024a; Shan et al. 2024). Furthermore, their self-attention mechanism, with its quadratic computational complexity scaling with image resolution, presents a significant challenge for processing high-resolution inputs. Consequently, we opt to use the more efficient Swin Transformer (Liu et al. 2021) as our visual encoder, which has been pre-trained on full-text recognition tasks (Kim et al. 2022). It enables us to process an entire image as a single input.

Dynamic Token Slimming

Although the APS module has been effective in removing some redundant regions, it is limited to processing entire rows or columns and is ineffective for more complex scenarios. In contrast, our Dynamic Token Slimming (DTS) method offers a more flexible approach to eliminate redundant visual content at the token level. The design of the DTS is based on the premise that a well-trained visual encoder should be capable of effectively distinguishing between essential and nonessential regions within an image, with the features corresponding to different regions exhibiting sufficient discriminative properties. We propose to separate these two types of tokens directly through clustering and then aggregate nonessential tokens into essential ones. We next detail the proposed DTS, which comprises two main processes: Dual-center K-Means Clustering and Similarity Weighted Aggregation.

Dual-center K-Means Clustering. To distinguish tokens into essential and nonessential types without introducing additional parameters, we apply K-Means clustering with two centers to the output sequence from the visual encoder, as illustrated in Fig 1. However, after obtaining the two clusters,

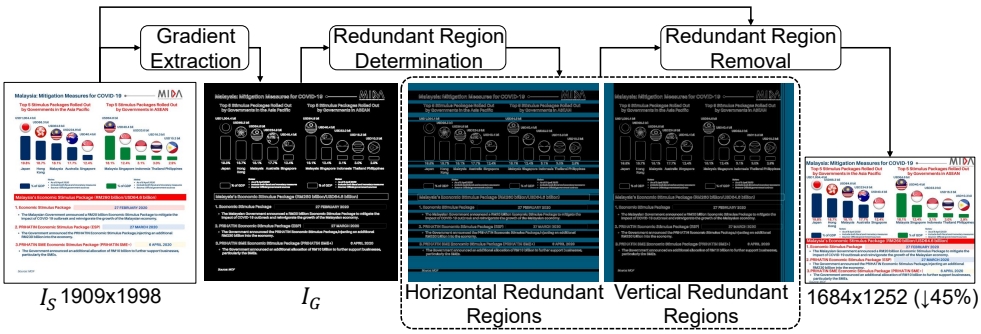

Figure 2: The proposed Adaptive Pixel Slimming module. It effectively reduces the resolution of document images by removing redundant regions.

Algorithm 1: Dual-center K-Means Clustering

Require:
visual tokens from Image Encoder: $V = \mathbb{R}^{L \times D}$
visual tokens from MLP Layer: $V' = \text{MLP}(V) \in \mathbb{R}^{L \times D'}$
CMX (calculate max similarity)
$R = 50$

cluster1, cluster2 $= K$-means$(V, \text{center_num} = 2)$
max_similarities = []
for token in V':
  max_similarities.append(CMX(token, other tokens in V'))
top-R_similar_tokens $=$ select_top_similar_tokens$(V', \text{max_similarities}, R)$
num1 =The number of common tokens between top-R_similar_tokens and cluster1
num2 =The number of common tokens between top-R_similar_tokens and cluster2
if num1 >num2:
  essential_tokens = cluster2
  nonessential_tokens = cluster1
else:
  essential_tokens = cluster1
  nonessential_tokens = cluster2
return essential_tokens, nonessential_tokens

it is unclear which cluster contains the essential tokens and which contains the non-essential tokens. Drawing on findings from (Liu et al. 2024d), nonessential tokens typically lack uniqueness and are often similar to other tokens. Therefore, we identify potential nonessential tokens by evaluating the similarity of each visual token with all other ones. The cluster with fewer of these nonessential tokens is determined to be the one containing the essential tokens. For more implementation details, please refer to Algorithm 1.

Similarity Weighted Aggregation. After obtaining essential and nonessential tokens, how to handle the nonessential tokens becomes a crucial issue. Inspired by (Kong et al. 2022; Liang et al. 2021; Wei et al. 2023), to mitigate potential clustering errors and prevent information loss, we do not discard the nonessential tokens outright. Instead, we aggregate them into the essential tokens to minimize the potential loss of effective information while enhancing the model's processing efficiency. As illustrated in Fig. 3, for each nonessential token from the clustering step, we find the most similar essential token. Each essential token may correspond to zero or multiple nonessential tokens. If an essential token has corresponding nonessential tokens, we perform a

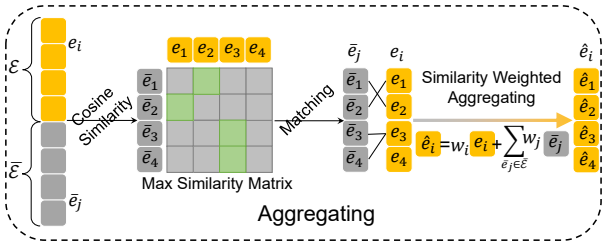

Figure 3: The proposed Similarity Weighted Aggregation module. It aggregates nonessential tokens into essential ones through similarity-weighted summation.

similarity-weighted summation of these tokens to obtain an aggregated token.

Specifically, after determining the set of essential tokens $\mathcal{E} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3, \ldots, \boldsymbol{e}_N\}$ and the set of nonessential tokens $\overline{\mathcal{E}} = \{\overline{\boldsymbol{e}}_1, \overline{\boldsymbol{e}}_2, \overline{\boldsymbol{e}}_3, \ldots, \overline{\boldsymbol{e}}_{\overline{N}}\}$, we aggregate nonessential tokens $\overline{\boldsymbol{e}}_j$ into essential tokens $\boldsymbol{e}_i$ by similarity weighted summation:

$$\hat{\boldsymbol{e}}_i = w_i \boldsymbol{e}_i + \sum_{\overline{\boldsymbol{e}}_j \in \overline{\mathcal{E}}} w_j \overline{\boldsymbol{e}}_j, \tag{1}$$

where w_i and w_j are the weight for the i-th essential and j-th nonessential token, and $\hat{\boldsymbol{e}}_i$ is the aggregated token. We use the similarity between e_i and $\overline{e}_j$ to determine w_i and w_j. Specifically, we first define the similarity matrix between essential tokens and nonessential tokens:

$$c_{i,j} = \frac{\boldsymbol{e}_i^T \overline{\boldsymbol{e}}_j}{\|\boldsymbol{e}_i\| \|\overline{\boldsymbol{e}}_j\|}. \tag{2}$$

Next, we obtain a mask matrix to ensure that each nonessential token is aggregated only with the most similar essential token:

$$m_{i,j} = \begin{cases} 1, & i = \arg\max_{\overline{\boldsymbol{e}}_j \in \overline{\mathcal{E}}} c_{i,j}, \\ 0, & \text{otherwise}, \end{cases} . \tag{3}$$

Finally, we can obtain aggregating weight w_j and w_i by:

$$w_j = \frac{\exp(c_{i,j}) m_{i,j}}{\sum_{\overline{\boldsymbol{e}}_j \in \overline{\mathcal{E}}} \exp(c_{i,j}) m_{i,j} + \text{e}} \tag{4}$$

Methods	Vis. Encoder (Params.)	Decoder	DocVQA^{val}	InfoVQA^{val}	ChartQA	FUNSD	SROIE	POIE
mPLUG-Owl (Ye et al. 2023b)	CLIP-L (0.3B)	LLaMA-7B	7.4	20.0	7.9	0.2	0.1	0.3
InstructBLIP (Dai et al. 2024)	CLIP-G (2B)	Vicuna-7B	4.5	16.4	5.3	0.2	0.6	1.0
LLaVA1.5 (Liu et al. 2024b)	CLIP-L (0.3B)	LLaMA-7B	8.5	14.7	9.3	0.2	1.7	2.5
Qwen-VL (Bai et al. 2023b)	CLIP-G (2B)	Qwen-7B	48.1	23.9	53.4	23.9	34.5	20.6
Monkey (Li et al. 2024b)	CLIP-G (2B)	Qwen-7B	50.1	25.8	54.0	24.1	41.9	19.9
InternVL (Chen et al. 2023b)	IViT (6B)	Vicuna-7B	28.7	23.6	45.6	6.5	26.4	25.9
InternLM-XComposer2 (Dong et al. 2024)	CLIP-L (0.3B)	InernLM2-7B	39.7	28.6	51.6	15.3	34.2	**49.3**
DocKylin (ours)	Swin (0.07B)	Qwen-7B	**65.1**	**34.8**	**59.0**	**25.5**	**49.5**	36.1

Table 1: Quantitative comparison of DocKylin with existing general-purpose MLLMs on document image benchmarks. Accuracy (%) proposed in (Zhang et al. 2024b) is adopted as the metric for all benchmarks.

Methods	Visual Encoder (Params.)	Decoder	DocVQA^{test}	InfoVQA^{test}	ChartQA	VisualMRC	DeepForm	WTQ
Donut (Kim et al. 2022)	Swin (0.07B)	BERT(1-4 layers)	67.5	11.6	41.8	93.9	61.6	18.8
Pix2Struct (Lee et al. 2023)	-	Transformers-1.3B	76.6	40.0	58.6	-	-	-
mPLUG-Doc (Ye et al. 2023b)	CLIP-L (0.3B)	LLaMA-7B	62.2	38.2	57.4	188.8	42.6	26.9
UReader (Ye et al. 2023a)	CLIP-L (0.3B)	LLaMA-7B	65.4	42.2	65.7	221.7	49.5	29.4
DocPeida (Feng et al. 2023a)	Swin (0.1B)	Vicuna-7B	47.1	15.2	46.9	-	-	-
HRVDA (Liu et al. 2024a)	Swin	LLaMA2-7B	72.1	43.5	**67.6**	211.5	63.2	31.2
Vary (Wei et al. 2025)	ViTDet-B & CLIP-L (0.4B)	Vicuna-7B	76.3	-	66.1	-	-	-
TextMonkey (Liu et al. 2024d)	CLIP-G (2B)	Qwen-7B	71.5	-	65.5	-	61.6	30.6
TextHawk (Yu et al. 2024)	SigLIP-SO (0.4B)	InternLM-7B	76.4	**50.6**	66.6	-	-	**34.7**
LayoutLLM (Luo et al. 2024)	LayoutLMv3-L (0.3B)	LLaMA2-7B	74.3	-	-	-	-	-
DocKylin (ours)	Swin (0.07B)	Qwen-7B	**77.3**	46.6	66.8	**319.9**	**64.2**	32.4

Table 2: Quantitative comparison of DocKylin with existing text-centric MLLMs on document-oriented benchmarks. The metrics used for each benchmark are consistent with those proposed in the original papers.

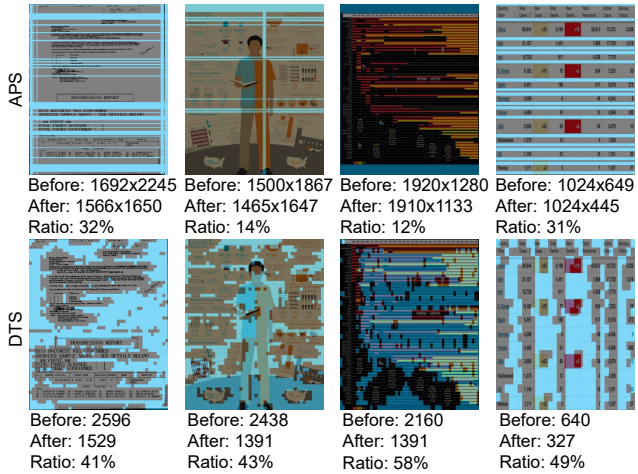

Figure 4: Visualization from DocKylin. The redundant pixels and nonessential tokens identified by Adaptive Pixel Slimming and Dynamic Token Slimming are highlighted in light blue. Zoom in for best view.

and

$$w_i = \frac{e}{\sum_{\overline{\boldsymbol{e}}_j \in \overline{\mathcal{E}}} \exp(c_{i,j}) m_{i,j} + \mathrm{e}}. \tag{5}$$

Experiments

Implementation Details

Our model underwent two training stages: pre-training and instruction tuning. All the training data are sourced from open-access datasets. Due to page limitations, specific datasets used are detailed in the supplementary materials.

During the pre-training stage, we trained only the visual encoder and the MLP, keeping the LLM parameters fixed. In this phase, the model learns to perform full-text recognition on document images and convert charts and tables into markdown formats (Hu et al. 2024). The goal is to enable the visual encoder to comprehend document content and align the visual features with the language space through the MLP.

During the pre-training phase, the model was trained for 450,000 iterations with a batch size of 8. The learning rate decayed from 1e-4 to 1e-5 using cosine annealing, which took approximately 40 A800 GPU days. In the instruction tuning phase, the model underwent 300,000 iterations with the same batch size and learning rate as in the pre-training phase. This phase further consumed approximately 24 A800 GPU days. Our Adaptive Pixel Slimming method is employed in both phases, while Dynamic Token Slimming (DTS) is used only during the instruction tuning phase. The maximum allowed number of pixels, MAX_SIZE, is set to 1728×1728. We initialize our visual encoder with Donut-Swin (0.07B) (Kim et al. 2022), pre-trained on the reading texts task, and use Qwen-7B-Chat (Bai et al. 2023a) as our LLM.

Comparisons with Prior Arts

We first compare our model with several existing general-purpose multimodal large language models (MLLMs). As these models sometimes tend to provide detailed responses, following (Zhang et al. 2024b), we adopt accuracy as our metric: a response is considered correct if it contains the complete answer. The results, as shown in Table 1, demonstrate that DocKylin achieves a significant advantage over the competing models. This highlights that VDU tasks remain challenging for current general-purpose MLLMs. Note

Model	Compression	Adaptive	DocVQA^{val}		InfoVQA^{val}		SROIE	
			Avg. len	Acc ↑	Avg. len	Acc ↑	Avg. len	Acc ↑
LLaVA1.5 (Liu et al. 2024b)	-	-	576	8.5	576	14.7	576	1.7
	Random	✗	300	5.4	300	14.3	300	0.5
	Random	✗	200	4.2	200	14.2	200	0
	MaxPool1D	✗	288	6.7	288	13.8	288	0.5
	AvgPool1D	✗	288	7.3	288	14.5	288	1.1
	FastV (K=2, R=50%) (Chen et al. 2025)	✗	306	8.4	306	14.4	306	2.0
	PruMerge (Shang et al. 2024)	✓	**123**	5.3	**123**	13.8	**121**	0.4
	PruMerge+ (Shang et al. 2024)	✓	232	5.2	233	14.0	229	0.3
	APS+DTS w/o Aggregation (ours)	✓	298	**9.9**	270	14.2	296	**3.4**
	APS+DTS (ours)	✓	298	**9.9**	270	**14.9**	296	3.3

Table 3: Comparisons with current SOTA parameter-free visual token compression methods. Accuracy proposed in (Zhang et al. 2024b) is adopted as the metric for all benchmarks. 'DTS w/o Aggregation' refers to skipping the Similarity Weighted Aggregation process mentioned earlier and directly discarding the nonessential tokens.

APS	DTS	Token Len.	Train	Infer.	DocVQA^{val}	ChartQA	InfoVQA	POIE
✗	✗	2770	1	1	63.1	55.8	**34.8**	34.5
✓	✗	2198	0.97	0.81	64.5	58.5	34.7	34.9
✓	✓	**1250**	**0.86**	**0.61**	**65.1**	**59.0**	**34.8**	**36.1**

Table 4: The training and inference time are normalized to the results in the first row. The token length and inference time are obtained based on the DocVQA^{val} dataset. Accuracy from (Zhang et al. 2024b) is adopted as the metric.

that descriptions of all these benchmarks are included in the supplementary materials.

We also compare our model with several existing text-centric MLLMs. The results, presented in Table 2, use the same metrics as those employed in the original benchmarks. Our method demonstrates advantages over these text-centric MLLMs, achieving leading performance across multiple benchmarks. Notably, our visual encoder is more lightweight than those used by the compared methods.

Some visual results from DocKylin on these benchmarks are presented in Fig. 4. The first row shows the results of redundant regions detected by APS, with corresponding text explaining the resulting reduction in resolution. The second row displays the nonessential tokens removed by DTS, with accompanying text illustrating its effect on compressing the visual sequence. These results demonstrate their effectiveness in accurately retaining important regions while significantly reducing the resolution and the length of the visual sequence.

To further demonstrate the superiority of our method, we compared it with other parameter-free visual token compression methods, using LLaVA1.5 (Liu et al. 2024b) as our baseline model (without retraining). The results in Table 3 indicate that our APS and DTS achieve performance improvements with shorter sequence lengths than the baseline model, significantly outperforming existing SOTA compression methods. We also found that incorporating weighted aggregation provided a slight performance boost. After model training, we expect enhanced benefits (Shang et al. 2024; Chen et al. 2025), as aggregation introduces an unfamiliar pattern to the model. Theoretically, it offers a higher performance ceiling by retaining more original information.

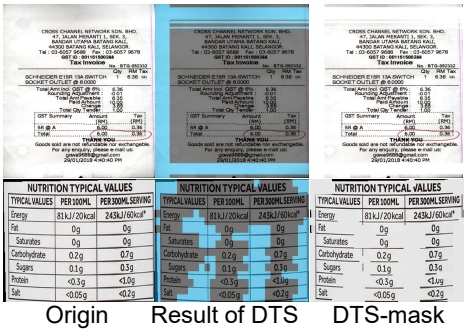

Figure 5: The results of APS-mask and DTS-mask. The identified redundant regions are highlighted in light blue. To minimize any additional effects caused by masking, the values in the masked regions are set to the average value of the pixels in the current region.

Ablation Studies

The Effectiveness of APS and DTS on DocKylin. We validate the effectiveness of APS and DTS on DocKylin and present the results in Table 4. Note that, settings without DTS in Table 4 refer to retraining a new model from scratch without employing DTS in the instruction tuning phase, rather than simply removing DTS from the already trained DocKylin model. The results demonstrate that both APS and DTS effectively reduce token sequence length and enhance computational efficiency, while also improving performance in most cases. This aligns with findings from previous studies (Liu et al. 2024d), which suggest that excessively long sequences due to high resolution may degrade performance. This degradation likely occurs because the visual sequences contain too many redundant tokens, placing greater demands on the model's ability to handle long contexts and making the learning more challenging. Furthermore, the APS and DTS methods can also be used during the training phase to accelerate training. Specifically, applying the DTS method during the instruction tuning phase can reduce training time by approximately 15%.

The accuracy of Identified Redundant Regions. We conduct experiments to further explore: **1)** How accurately do APS and DTS identify redundant regions? **2)** Do APS

Strategy	Scanned Document		Infographics		Chart		Table		Receipt		Ingredient List	
	Mask Ratio	Acc.↑	Mask Ratio	Acc.↑	Mask Ratio	Acc.↑	Mask Ratio	Acc.↑	Mask Ratio	Acc.↑	Mask Ratio	Acc.↑
None	-	55.2	-	26.7	-	73.5	-	34.8	-	44.0	-	25.9
APS-R-mask	63.4	0.2	83.3	14.4	68.5	3.8	96.0	4.8	36.1	0.1	99.4	0.1
DTS-R-mask	51.8	0.2	64.8	14.6	57.6	4.4	61.9	0.6	55.9	0.1	66.0	0.5
Top-30-mask	30	50.4	30	24.0	30	74.3	30	25.8	30	37.1	30	19.7
Top-40-mask	40	45.3	40	22.3	40	71.0	40	23.1	40	32.3	40	15.4
APS-mask	36.6	54.7	16.7	26.4	31.5	74.8	4.0	34.6	63.9	44.0	0.6	26.1
DTS-mask	48.2	54.9	35.2	26.3	42.4	73.8	38.1	31.5	44.1	43.2	34.0	25.1

Table 5: The Scanned Document, Infographics, Chart, Table, Webpage, Receipt, and Ingredient List are sourced from the DocVQA^{val}, InfoVQA^{val}, ChartQA (augmented), WTQ, visualMRC, SROIE and POIE datasets, respectively. All data, except for the Webpage which uses the CIDEr metric, are evaluated using the accuracy metric as proposed in (Zhang et al. 2024b).

Methods	Resolution	DocVQA^{val}	InfoVQA^{val}	SROIE	FUNSD
LLaVA1.5 (Liu et al. 2024b)	336×336	8.5	14.7	1.7	0.2
LLaVA1.5 (Liu et al. 2024b) + APS		10.7 (**+27.4%**)	14.7 (**+0%**)	3.7 (**+118%**)	0.92 (**+360%**)
Qwen-VL (Bai et al. 2023b)	448×448	48.1	23.9	34.5	20.6
Qwen-VL (Bai et al. 2023b) + APS		51.2 (**+6.4%**)	24.7 (**+4.1%**)	40.0 (**+15.9%**)	24.3 (**+17.9%**)
Monkey (Li et al. 2024b)	896×896	50.1	25.8	41.9	24.1
Monkey (Li et al. 2024b) + APS		56.3 (**+12.4%**)	27.5 (**+6.6%**)	47.0 (**+12.2%**)	27.3 (**+13.3%**)
InternVL2 (Chen et al. 2024)	448×448×(1∼12)	76.2	49.5	54.7	41.7
InternVL2 (Chen et al. 2024) + APS		76.1	48.2	54.2	40.6
InternVL2 (Chen et al. 2024) + APS + Resize		77.3 (**+1.4%**)	49.4 (**-0.2%**)	55.2 (**+0.9%**)	43.4 (**+4.1%**)

Table 6: The enhancement brought to other MLLMs by Adaptive Pixel Slimming. **Relative improvements** are highlighted in bold. Accuracy proposed in (Zhang et al. 2024b) is adopted as the metric for all benchmarks.

and DTS demonstrate consistent performance across different data types? Specifically, we apply APS and DTS to identify redundant regions in different document images, and then mask these regions instead of removing them (as shown in Fig. 5). The masked images are used as inputs to an existing MLLM to assess any performance degradation. This approach allows us to isolate the accuracy of APS and DTS from the performance improvements brought by redundancy removal.

Table 5 shows that APS effectively reduces resolution across all image types with minimal performance impact. However, the reduction effects on the table and ingredient list are less pronounced, likely due to the lines spanning the entire image, making APS difficult to identify redundant rows or columns. DTS also effectively reduces resolution across all image types and typically achieves greater resolution reduction than APS, as it can identify local areas beyond entire rows and columns. While DTS may cause noticeable performance drops in the table datasets, our visual analysis suggests this is due to new patterns introduced by masking table lines, not the masking of critical text content. Practically, such potential performance degradation from DTS can be largely mitigated: **1)** DTS retains information by similarity-weighted aggregation approach rather than completely discarding redundant tokens; and **2)** during image encoding, information from masked areas is already integrated into neighboring tokens.

We conduct additional experiments by reversing the APS-mask and DTS-mask results by masking the non-redundant regions identified by APS and DTS, denoted as APS-R-mask and DTS-R-mask. The results indicate a significant performance drop, further demonstrating the accuracy of APS and DTS in identifying redundant regions.

We further compare our approach with methods that directly select redundant tokens based on similarity priors (e.g., TextMonkey and PruMerge adopt this approach). Specifically, in Table 5, Top-30-mask and Top-40-mask represent masking the top 30% and 40% of tokens with the highest similarity to other tokens, respectively, as nonessential tokens. Results show that DTS outperforms the top K strategy, even at higher reduction rates.

Potentials for Further Application and Research. Since APS is a parameter-free method that requires no additional training or modification of model architecture, it can be easily applied to other MLLMs to enhance their performance on high-resolution document images. To further demonstrate its effectiveness, we apply APS to various existing MLLMs (employing it before the image processing module of these models). As shown in Table 6, APS consistently improves the performance of MLLMs that support lower resolutions (LLaVA1.5, Qwen-VL and Monkey). However, directly applying APS to high-resolution models like InternVL2 results in a slight performance decline, likely due to a mismatch between the dense text produced by APS and the high-resolution training setup of InternVL2. Adjusting the APS-processed images back to their original resolution (InternVL2+APS+Resize) can achieve a performance gain of up to 4.1%. This suggests that APS alone is most effective for low-resolution models, while APS+Resize is more suitable for high-resolution models.

The redundant regions in document images are often simple, such as plain backgrounds, distinguishing these regions is relatively easy. This raises the question of whether DTS's effectiveness is due to this simplicity. Although this paper primarily explores VDU tasks, we are also interested in the generalizability of DTS. Therefore, we conducted some experiments on scene text images (Tang et al. 2022; Liu et al. 2023b; Zhang et al. 2020). Specifically, we retrained a scene text image full-text recognition model using some open-source scene text data, performing only the first pre-training

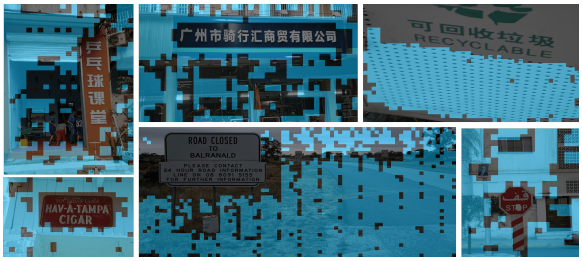

Figure 6: Effects of applying DTS to natural scene images.

phase. We then applied DTS to the trained model and tested it on the TextVQA dataset. Some visual results are shown in Fig. 6. It can be seen that DTS is effective in distinguishing nonessential regions even in the complex backgrounds of natural scene images.

Conclusions

To address the issues of high input resolution and excessively long token sequences in current MLLMs when counters with VDU tasks, we propose the DocKylin model. It includes an Adaptive Pixel Slimming (APS) preprocessing module that removes redundant regions from document images, effectively reducing their resolution. The model employs a lightweight and flexible image encoding approach to adapt to the varying resolutions and aspect ratios of document image scenarios. Moreover, we propose a token compression module, Dynamic Token Slimming (DTS), which clusters and selects essential tokens from a long visual sequence and employs aggregation to avoid potential information loss. Experiments demonstrate that DocKylin achieves leading performance across multiple VDU benchmarks. Extensive experiments demonstrate that APS and DTS effectively reduce the length of visual tokens, enhancing both computational efficiency and model performance. The parameter-free nature of our APS and DTS modules not only enhances efficiency but also facilitates easy integration into existing multimodal large language models, and our experiments indicate their potential for broader applications and scenarios. This feature makes our approach particularly valuable for improving document understanding capabilities in a wide range of applications.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China (Grant No.: 62441604, 62476093).

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. 2023a. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023b. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Cai, M.; Yang, J.; Gao, J.; and Lee, Y. J. 2024. Matryoshka Multimodal Models. *arXiv preprint arXiv:2405.17430*.

Cha, J.; Kang, W.; Mun, J.; and Roh, B. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, 13817–13827.

Chen, J.; Zhu, D.; Shen, X.; Li, X.; Liu, Z.; Zhang, P.; Krishnamoorthi, R.; Chandra, V.; Xiong, Y.; and Elhoseiny, M. 2023a. MiniGPT-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2025. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*, 19–35.

Chen, Z.; Wang, W.; Tian, H.; Ye, S.; Gao, Z.; Cui, E.; Tong, W.; Hu, K.; Luo, J.; Ma, Z.; et al. 2024. How Far Are We to GPT-4V? Closing the Gap to Commercial Multimodal Models with Open-Source Suites. *arXiv preprint arXiv:2404.16821*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Muyan, Z.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 2023b. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*.

Cheng, H.; Zhang, P.; Wu, S.; Zhang, J.; Zhu, Q.; Xie, Z.; Li, J.; Ding, K.; and Jin, L. 2023. M6Doc: A large-scale multi-format, multi-type, multi-layout, multi-language, multi-annotation category dataset for modern document layout analysis. In *CVPR*, 15138–15147.

Cui, L.; Xu, Y.; Lv, T.; and Wei, F. 2021. Document AI: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2024. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *NeurIPS*, 36.

Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Wang, B.; Ouyang, L.; Zhang, S.; Duan, H.; Zhang, W.; Li, Y.; et al. 2024. InternLM-XComposer2-4KHD: A Pioneering Large Vision-Language Model Handling Resolutions from 336 Pixels to 4K HD. *arXiv preprint arXiv:2404.06512*.

Feng, H.; Liu, Q.; Liu, H.; Zhou, W.; Li, H.; and Huang, C. 2023a. DocPedia: Unleashing the Power of Large Multimodal Model in the Frequency Domain for Versatile Document Understanding. *arXiv preprint arXiv:2311.11810*.

Feng, H.; Wang, Z.; Tang, J.; Lu, J.; Zhou, W.; Li, H.; and Huang, C. 2023b. UniDoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*.

Fu, C.; Zhang, Y.-F.; Yin, S.; Li, B.; Fang, X.; Zhao, S.; Duan, H.; Sun, X.; Liu, Z.; Wang, L.; et al. 2024. MME-Survey: A Comprehensive Survey on Evaluation of Multimodal LLMs. *arXiv preprint arXiv:2411.15296*.

Hu, A.; Xu, H.; Ye, J.; Yan, M.; Zhang, L.; Zhang, B.; Li, C.; Zhang, J.; Jin, Q.; Huang, F.; et al. 2024. mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. *arXiv preprint arXiv:2403.12895*.

Huang, M.; Liu, Y.; Liang, D.; Jin, L.; and Bai, X. 2024. Mini-Monkey: Alleviating the Semantic Sawtooth Effect for Lightweight MLLMs via Complementary Image Pyramid. *arXiv preprint arXiv:2408.02034*.

Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; and Park, S. 2022. OCR-free document understanding transformer. In *ECCV*, 498–517.

Kim, G.; Lee, H.; Kim, D.; Jung, H.; Park, S.; Kim, Y.; Yun, S.; Kil, T.; Lee, B.; and Park, S. 2023. Visually-Situated Natural Language Understanding with Contrastive Reading Model and Frozen Large Language Models. In *EMNLP*.

Kong, Z.; Dong, P.; Ma, X.; Meng, X.; Niu, W.; Sun, M.; Shen, X.; Yuan, G.; Ren, B.; Tang, H.; et al. 2022. SPViT: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, 620–640.

Lee, K.; Joshi, M.; Turc, I. R.; Hu, H.; Liu, F.; Eisenschlos, J. M.; Khandelwal, U.; Shaw, P.; Chang, M.-W.; and Toutanova, K. 2023. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *ICML*, 18893–18912.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 19730–19742.

Li, X.; Wu, Y.; Jiang, X.; Guo, Z.; Gong, M.; Cao, H.; Liu, Y.; Jiang, D.; and Sun, X. 2024a. Enhancing visual document understanding with contrastive learning in large visual-language models. In *CVPR*, 15546–15555.

Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; and Bai, X. 2024b. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*, 26763–26773.

Liang, Y.; Chongjian, G.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2021. EViT: Expediting Vision Transformers via Token Reorganizations. In *ICLR*.

Liao, W.; Wang, J.; Li, H.; Wang, C.; Huang, J.; and Jin, L. 2024. DocLayLLM: An Efficient and Effective Multi-modal Extension of Large Language Models for Text-rich Document Understanding. *arXiv preprint arXiv:2408.15045*.

Lin, Z.; Lin, M.; Lin, L.; and Ji, R. 2024. Boosting Multimodal Large Language Models with Visual Tokens Withdrawal for Rapid Inference. *arXiv preprint arXiv:2405.05803*.

Lin, Z.; Liu, C.; Zhang, R.; Gao, P.; Qiu, L.; Xiao, H.; Qiu, H.; Lin, C.; Shao, W.; Chen, K.; et al. 2023. SPHINX: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*.

Liu, C.; Yin, K.; Cao, H.; Jiang, X.; Li, X.; Liu, Y.; Jiang, D.; Sun, X.; and Xu, L. 2024a. HRVDA: High-resolution visual document assistant. In *CVPR*, 15534–15545.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024b. Improved baselines with visual instruction tuning. In *CVPR*, 26296–26306.

Liu, H.; You, Q.; Han, X.; Liu, Y.; Huang, H.; He, R.; and Yang, H. 2024c. Visual Anchors Are Strong Information Aggregators For Multimodal Large Language Model. *arXiv preprint arXiv:2405.17815*.

Liu, Y.; Li, Z.; Li, H.; Yu, W.; Huang, M.; Peng, D.; Liu, M.; Chen, M.; Li, C.; Jin, L.; et al. 2023a. On the hidden mystery of OCR in large multimodal models. *arXiv preprint arXiv:2305.07895*.

Liu, Y.; Yang, B.; Liu, Q.; Li, Z.; Ma, Z.; Zhang, S.; and Bai, X. 2024d. TextMonkey: An OCR-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

Liu, Y.; Zhang, J.; Peng, D.; Huang, M.; Wang, X.; Tang, J.; Huang, C.; Lin, D.; Shen, C.; Bai, X.; et al. 2023b. SPTS v2: single-point scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin Transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Lu, J.; Yu, H.; Wang, Y.; Ye, Y.; Tang, J.; Yang, Z.; Wu, B.; Liu, Q.; Feng, H.; Wang, H.; et al. 2024. A bounding box is worth one token: Interleaving layout and text in a large language model for document understanding. *arXiv preprint arXiv:2407.01976*.

Luo, C.; Shen, Y.; Zhu, Z.; Zheng, Q.; Yu, Z.; and Yao, C. 2024. LayoutLLM: Layout Instruction Tuning with Large Language Models for Document Understanding. In *CVPR*, 15630–15640.

OpenAI. 2022. ChatGPT: Get answers. Find inspiration. Be more productive. https://openai.com/chatgpt/. Accessed: 2024-05-10.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Shan, B.; Fei, X.; Shi, W.; Wang, A.-L.; Tang, G.; Liao, L.; Tang, J.; Bai, X.; and Huang, C. 2024. MCTBench: Multimodal cognition towards text-rich visual scenes benchmark. *arXiv preprint arXiv:2410.11538*.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Tanaka, R.; Iki, T.; Nishida, K.; Saito, K.; and Suzuki, J. 2024. InstructDoc: A dataset for zero-shot generalization of visual document understanding with instructions. In *AAAI*, volume 38, 19071–19079.

Tang, J.; Lin, C.; Zhao, Z.; Wei, S.; Wu, B.; Liu, Q.; Feng, H.; Li, Y.; Wang, S.; Liao, L.; et al. 2024a. TextSquare: Scaling up Text-Centric Visual Instruction Tuning. *arXiv preprint arXiv:2404.12803*.

Tang, J.; Liu, Q.; Ye, Y.; Lu, J.; Wei, S.; Lin, C.; Li, W.; Mahmood, M. F. F. B.; Feng, H.; Zhao, Z.; et al. 2024b. MTVQA: Benchmarking Multilingual Text-Centric Visual Question Answering. *arXiv preprint arXiv:2405.11985*.

Tang, J.; Zhang, W.; Liu, H.; Yang, M.; Jiang, B.; Hu, G.; and Bai, X. 2022. Few could be better than all: Feature sampling and grouping for scene text detection. In *CVPR*, 4563–4572.

The InternLM Team. 2023. InternLM: A multilingual language model with progressively enhanced capabilities.

The Vicuna Team. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. Accessed: 2024-05-10.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, A.-L.; Shan, B.; Shi, W.; Lin, K.-Y.; Fei, X.; Tang, G.; Liao, L.; Tang, J.; Huang, C.; and Zheng, W.-S. 2024. Pargo: Bridging vision-language with partial and global views. *arXiv preprint arXiv:2408.12928*.

Wang, Y.; Zhou, W.; Feng, H.; Zhou, K.; and Li, H. 2023. Towards improving document understanding: An exploration on text-grounding via MLLMs. *arXiv preprint arXiv:2311.13194*.

Wei, H.; Kong, L.; Chen, J.; Zhao, L.; Ge, Z.; Yang, J.; Sun, J.; Han, C.; and Zhang, X. 2025. Vary: Scaling up the vision vocabulary for large vision-language model. In *ECCV*, 408–424.

Wei, S.; Ye, T.; Zhang, S.; Tang, Y.; and Liang, J. 2023. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *CVPR*, 2092–2101.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *ACM SIGKDD*, 1192–1200.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Ye, J.; Hu, A.; Xu, H.; Ye, Q.; Yan, M.; Xu, G.; Li, C.; Tian, J.; Qian, Q.; Zhang, J.; et al. 2023a. UReader: Universal OCR-free Visually-situated Language Understanding with Multimodal Large Language Model. In *EMNLP*.

Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. 2023b. mPLUG-Owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Yin, S.; Fu, C.; Zhao, S.; Li, K.; Sun, X.; Xu, T.; and Chen, E. 2024. A survey on multimodal large language models. *National Science Review*.

Yu, Y.-Q.; Liao, M.; Wu, J.; Liao, Y.; Zheng, X.; and Zeng, W. 2024. TextHawk: Exploring Efficient Fine-Grained Perception of Multimodal Large Language Models. *arXiv preprint arXiv:2404.09204*.

Zhang, J.; Chen, B.; Cheng, H.; Guo, F.; Ding, K.; and Jin, L. 2023a. DocAligner: Annotating real-world photographic document images by simply taking pictures. *arXiv preprint arXiv:2306.05749*.

Zhang, J.; Liang, L.; Ding, K.; Guo, F.; and Jin, L. 2023b. Appearance enhancement for camera-captured document images in the wild. *IEEE Transactions on Artificial Intelligence*.

Zhang, J.; Luo, C.; Jin, L.; Guo, F.; and Ding, K. 2022. Marior: Margin Removal and Iterative Content Rectification for Document Dewarping in the Wild. In *ACM MM*, 2805–2815.

Zhang, J.; Luo, C.; Jin, L.; Wang, T.; Li, Z.; and Zhou, W. 2020. SaHAN: Scale-aware hierarchical attention network for scene text recognition. *Pattern Recognition Letters*, 136: 205–211.

Zhang, J.; Peng, D.; Liu, C.; Zhang, P.; and Jin, L. 2024a. DocRes: A Generalist Model Toward Unifying Document Image Restoration Tasks. In *CVPR*, 15654–15664.

Zhang, S.; Yang, B.; Li, Z.; Ma, Z.; Liu, Y.; and Bai, X. 2024b. Exploring the Capabilities of Large Multimodal Models on Dense Text. In *ICDAR*, 281–298.

Zhao, W.; Feng, H.; Liu, Q.; Tang, J.; Wei, S.; Wu, B.; Liao, L.; Ye, Y.; Liu, H.; Zhou, W.; et al. 2024a. TabPedia: Towards comprehensive visual table understanding with concept synergy. *NeurIPS*.

Zhao, Z.; Tang, J.; Wu, B.; Lin, C.; Wei, S.; Liu, H.; Tan, X.; Zhang, Z.; Huang, C.; and Xie, Y. 2024b. Harmonizing Visual Text Comprehension and Generation. *NeurIPS*.